Chapter 9: Installation and Maintenance

Overview

Below both installation and maintenance of TART95 are discussed. The major emphasis in this chapter will be on installation. The TART95 system, including TART95, its binary data files, and all of its related codes are now available for immediately use on a variety of computers, and emphasis here is on how to install and immediately use this system. Maintenance will only be discussed in passing, as it relates to those users who have to compile the FORTRAN source codes, create the binary data files, or resize TART95 for use on any individual computer.

Types of Computers

TART95 has been installed and used on the following types of computers,

CRAY (XMP and YMP)
SUN
SGI
HP
MEIKO
DEC-ALPHA
IBM-RSIC
IBM-PC

Others

TART95 is now so computer independent that it was very easy to implement on each of the above computers, and it should be fairly easy to implement on any other type of computer; see the below sections on computer dependence. If you have a different type of computer we would be interested in helping you to implement TART95 on your computer, e.g., feel free to contact the authors for assistance.

Availability

TART95 is available from the Radiation Shielding Information Center (RSIC), Oak Ridge, TN, and from the Nuclear Energy Agency/Data Bank (NEA/DB), Paris, France.

The TART95 system is available in two different forms: 1) for the computers named above, the programs are supplied as executables and the data files as binary, random access files. In this form all you need do is copy the files for your type of computer and start using them, 2) for other types of computers the FORTRAN source codes and data files in character form are supplied. In this form you have to compile the programs and create the binary, random access files before you can start using them.

The complete TART95 system is available on a 4 mm TAR tape. The TAR tape has a number of directories, one for each type of computer that TART95 has been implemented on. The directories have obvious names,

CRAY
SUN
SGI
HP
MEIKO
DECALPHA
IBMRSIC

For the convenience of IBM-PC users who probably do not have access to a 4 mm TAR disk drive, the all of the files are available on a set of four 3.5" diskettes that can be directly read by an IBM-PC.

Within each of these directories are a number of sub-directories. The names of these sub-directories are the same for each type of computer:

tart95	- TART95, its four binary data files, and an example input
tartchek	- TARTCHEK, its associated files, and an example input
utility	- CRITEDIT and FLUXEDIT
examples	- a variety of example TART95 input decks

All of the programs are in executable form, ready for use.

In addition to the directories for the above named types of computers, there is an additional directory for use on any other type of computer; it is named,

OTHER

This directory contains all of the FORTRAN source codes, and the four character files that are used to create the four binary, random access data files used by TART95. If you are using one of the above named computers and the distributed executables work on your computer you probably aren't interested in these files. Within this directory are a number of sub-directories,

tart95	- TART95 source and Makefile for a variety of computers
tartchek	- TARTCHEK source and Makefiles for a variety of computers
utility	- CRITEDIT and FLUXEDIT sources
makelibe	- four character files and utility codes to create binary files

Easy Installation

For use on any of the above named computers we suggest that you create a directory named tart95 on your hard drive, and within this directory copy the appropriate directory and all of its sub-directories from the 4 mm TAR tape to your hard drive. For example, for use on a SUN workstation, you need only copy the directory SUN. For use on an IBM-PC follow the same procedure to create a directory named tart95 and within this directory follow the instructions to copy the four 3.5" diskettes and install all of the programs.

All of the programs are in executable form, ready for use. You should immediately verify that the codes run on your computer; because the operating systems keep changing we cannot guarantee that the executable files supplied will run on your computer. To verify that they run try running TART95 and TARTCHEK. If they run installation is finished.

If they do not run you will have to re-compile the program; to do this follow the below instructions for **Other Installation**.

On an IBM-PC if the executable program does not run, you may be able to get the executable to run without re-compiling, by maximizing the available memory on your computer, as described below under the section on memory allocation. If you do need to re-compile on an IBM-PC you will need the ABSOFT FORTRAN-77 compiler.

Other Installation

There are two situations in which you may need the files from the OTHER directory: 1) you are using one of the above named computers, but the programs do not run on your computer, so you have to re-compile them. In this case you can still use the distributed binary, random access files; you do not have to create them again, 2) you are using a different type of computer, in which case you must both compile all of the codes and create the four binary, random access files used by TART95.

Warning - note, that the sub-directory names within the directories for each of the above named computers are the same as those in the OTHER directory. Therefore if you are going to use the OTHER directory, we suggest that within your TART95 directory you create a sub-directory named **sources** (since the contents of these files is mostly FORTRAN source code) and within this sub-directory copy all of the OTHER files. This will avoid all filename conflicts and prevent any of the files in your basic TART95 directory from being overwritten.

Installing TART95

For each type of computer the program is distributed with a Makefile, that will compile and load the code. All computer dependent coding is in the portion of the source code named tartlib, with an appropriate name for each type of computer, e.g., tartlib.SUN.f, tartlib.SGI.f, etc. Since IBM-PCs do not allow longer filenames, its file is simply named tartlib.f. The Makefile for each computer will automatically read the corresponding tartlib file for compilation and loading.

For use on other computers, you may have to supply routines to define time and date in character form and running time in floating point form (see, the section of this chapter on computer dependence for details). In order to do this you need merely replace a few of the routines from tartlib.f by your local equivalent routines. For all the computers we have used this has been very easy and straightforward. See, the differences between the various tartlib.f files to see how these are defined on various computers. If all else fails and you cannot find routines to define time, date, and running time, you can always add simple dummy routines that return either blanks or a fixed time and date, or zero for running time. This will allow you to get TART95 running and you can return to this problem later.

Installing TARTCHEK

To install TARTCHEK, within the directory OTHER directory and TARTCHEK sub-directory run the Makefile. Interactive graphics are very computer dependent and running Makefile may not produce an executable. Having installed this code on many different workstations the only difficulty has been locating where the UNIX Xlib graphics library is located on each system. Locating Xlib usually requires minor changes to the Makefile to correctly define the path command to this library. This directory contains Makefiles for a variety of computers so that you can see the different paths to Xlib on different workstations. Try several of them and if all else fails consult your system supervisor. This computer dependent location of Xlib on UNIX systems is a problem, but hang in there and keep trying, because until now we haven't run into any computer where we weren't able to locate Xlib after a few tries.

Installing the Utility Codes

The utility codes CRITEDIT and FLUXEDIT are extremely simple, and in completely standard FORTRAN that will run on virtually any computer. Within the OTHER directory and utility sub-directory, compile the two codes. They are completely self contained and run so fast that optimization or any other special options are not required. To compile the two codes critedit.f and fluxedit.f and create the executables critedit.exe and fluxedit.exe type:

```
f77 -o critedit.exe critedit.f

f77 -o fluxedit.exe fluxedit.f
```

Creating Binary Data Files

TART95 is distributed with a series of relatively simple standard FORTRAN codes that can be used to create the four binary, random access data files for use on any computer. On any given computer these codes need only be run once, to initially create the binary data files. Once the binary files have been created the character files are no longer required and they can be discarded. See the chapter on **Utility Codes** for instructions on how to create the four binary, random access files. See also the below section on **Binary Data File Computer Dependence**

The TART95 binary data files are supplied for use on all of the computers that TART95 has been implemented on; this includes: CRAY, HP, SUN, SGI, Meiko, DEC-Alpha, IBM-RSIC and IBM-PC. For these computers you will not be concerned with the following utility codes.

If you wish to use TART95 on any other type of computer you must first create the binary data files. For use on any computer the four data files are distributed in character form, compatible to be read on any computer. The four character files are,

TARTND.LST	- Neutron interaction data
TARTPPD.LST	- Neutron induced photon production data
EGDL.DAT	- Photon interaction data
CROSS.LST	- Neutron self-shielding, multi-band parameter data

Each of these is read by a simple FORTRAN code, and converted to binary, random access files for use by TART95. The four codes used are,

;	process neutron interaction data
	process neutron induced photon production data
	process photon interaction data
	process neutron self-shielding data

The four binary, random access files produced for use by TART95 are:

TARTND	- Neutron interaction data
TARTPPD	- Neutron induced photon production data
GAMDAT	- Photon interaction data
NEWCROSS	- Neutron self-shielding, multi-band parameter data

To create the four files, compile the four programs and run them one after another. When they have all been run, move the four binary, random access files to your directory containing the executable version of TART95. You only need to create these files once. Therefore you can now delete the character files and four codes.

Warning - The only computer dependent coding in these four codes is the definition of the record length for the binary, random access files. On all computers these files are written in fixed length records, with each record 10,000 words long. On most computers the record length in the open statement is in bytes. However, on SGI and DEC-Alpha it is in words. For a CRAY the record length MUST be 80,000 (10,000 64 bit words). On most workstations and IBM-PC it MUST be 40,000 (10,000 32 bit words). On SGI and DEC-Alpha it MUST be 10,000 (10,000 32 bit words). As distributed in these four codes the record length is defined to be 40,000. If you are using a different type of computer we recommend that you first try running tartnd with this record length. If the resulting TARTND binary, random access file is 2,000,000 bytes long it has been created correctly. If it is 8,000,000, this indicates that on your computer the record length is defined in words, not bytes. In this case, in all four codes change 40000 to 10000 throughout, and then run the codes; TARTND should then be 2,000,000 bytes long. If tartnd does not run to completion and instead dies with a message that the output exceeds the defined record length, you are probably using a 64 bit per word computer and should change 40000 to 80000 through the four codes and then run the codes.

Other Considerations

User Feedback

The most important improvements in codes, such as TART95, are the result of feedback from users. No code is perfect, but any code can be improved by pooling the experience of all users. Therefore we ask you, the users, to inform us if you have any suggestions how TART95 can be improved to better need your needs.

Of particular interest to us is implementation of TART95 are computers others than those described above, that TART95 has already been implemented on. It would be of great help to us, and all other users, if you could supply us with information concerning how you implemented TART95. If you do supply us with this information it will also minimize the work that you have to do in the future, since we will than maintain a version of TART95 for your computer and supply you with future versions of the code that should immediately run on your computer.

In order to maintain reasonable standard to minimize maintenance we ask you to,

1) Only make those changes that are absolutely required to implement the code. All of the computer dependent aspects of TART95, described below, are contained in tartlib for each computer. Therefore if you must make changes to implement TART95, only make changes to tartlib - if at all possible do not change any other portions of the code.

2) Send us your Makefile and tartlib. We will maintain them and distribute updated copies whenever TART95 is modified; this will minimize the amount of work that you have to do in the future.

FORTRAN Computer Dependence

TART95 is 100 % FORTRAN-77. No special C, or other language, routines are required and there are no special subroutine or function libraries required. To use TART95 you need merely compile it using the appropriate Makefile for your computer and then start using it. The TART95 system is distributed with Makefiles for all of the computers that it has been used on, e.g., there are files named Makefile.SUN, Makefile.SGI, etc. All of the computer dependent portions of the code are in tartlib, which is supplied for each computer and properly referenced by the corresponding Makefile, e.g., Makefile.SUN will use tartlib.SUN.f, and Makefile.SGI will use tartlib.SGI.f.

As a starting point to implement TART95 on any other type of computer, we suggest that you initially use Makefile.SUN. Based on your compiler results you can then decide whether or not you have any of the computer dependent problems, described below. If you do, it is usually quite simple to modify tartlib to meet your needs.

The only FORTRAN computer dependent portions of the TART95 code are associated with,

1) TART95 uses pointers to dynamically assign memory to arrays. On CRAY, most workstations and IBM-PC these can be treated as normal integer variables (64 bits on CRAY, 32 bits on most workstations and IBM-PC). The exception is that on DEC-Alpha pointers are 64 bit variables and care must be used to maintain this 64 bit representation. See, subroutine MEMLOW, in tartlib, to see how this problem is dealt with on various computers.

2) How to read parameters from the execution line, to allow users to define the input and output listing filenames. See, subroutine INLINE, in tartlib, to see how this problem is dealt with on various computers.

3) How to define the date and time of execution in character form. See subroutine DATEIT, in tartlib, to see how this problem is dealt with on various computers.

4) How to define running time in seconds. See subroutine TIMEIT, in tartlib, to see how this problem is dealt with on various computers.

5) The record length of the binary data files is defined in **bytes** on CRAY, most workstations and IBM-PC. On SGI and DEC-Alpha the record length of the binary files is defined in **words**. On all computers the binary data files are in fixed length records, where each record is 10,000 words. On CRAY this means the record length is 80,000 bytes. On most workstations and IBM-PC it is 40,000 bytes. On SGI and DEC-Alpha it is 10,000 words. See subroutines TARTOPEN, RD64OPN and WR64OPN, in tartlib, to see how this problem is dealt with on various computers.

6) TART95 uses **REAL*8** arithmetic throughout; this means **DOUBLE PRECISION** on all workstations and IBM-PCs, but **SINGLE PRECISION** on CRAY computers. For use on CRAY computers TART95 is first run through a pre-processor to change all DOUBLE PRECISION function calls to SINGLE PRECISION calls, e.g., convert DCOS to COS. Otherwise the CRAY version of TART95 is completely compatible with the workstation version of the code.

These are the only FORTRAN computer dependent portions of TART95. All computer dependent coding is in the tartlib portion of the code, and separate versions of tartlib are distributed for each type of computer. For example, there are files named tartlib.SUN.f, tartlib.SGI.f, etc., and the correct file is defined to be loaded on each computer by using the correct Makefile, e.g., Makefile.SUN, Makefile.SGI, etc.

Binary Data File Computer Dependence

There are four data files used by TART95 to handle: 1) neutron interaction data (**TARTND**), 2) Photon interaction data (**GAMDAT**), 3) neutron induced photon production (**TARTPPD**), and 4) neutron self-shielding data (**NEWCROSS**). The contents and formats of these files are described elsewhere in this report under the chapter on **Binary Data File Formats**.

The TART95 binary data files are designed to be used on virtually any computer. The processed data files are written in character form that can easily be copied to any computer, where they are then converted to binary, random access form, compatible for use on that computer. On any computer the files are created and accessed using standard FORTRAN binary, random access statements that work on any computer. This operation has now been successfully performed on every computer from an IBM-PC to a CRAY super computer. As such, it is not accurate to say that the files are binary compatible between computers, e.g., the files prepared for use on a SUN workstation cannot be used on an SGI workstation. However, what is important is that these binary, random access files can be accurately implemented and efficiently used on virtually any computer. See the chapter on **Utility Codes** for instructions on how to create these files.

Memory Requirements

As distributed, the production version on TART95 for use on workstations requires 32 megabytes of memory. For use on IBM-PCs, three versions are distributed for use on 8, 16 or 32 megabyte PCs.

The code itself is quite small; usually only about 800 kilobytes on most computers. The bulk of the memory used by the code is used for dynamic memory allocation using POINTERS. The code does not use true dynamic memory allocation. It has a fixed sized blank common array named **tarheap** and portions of this array are equated to other arrays as needed using POINTERS.

As distributed, the production version of the code assigns 4 million 8 byte words to **tarheap** (32 megabytes), which is more than enough to run virtually any TART95 problem. The parameter defining the size of **tarheap** (**MAXCORE**) is defined in the INCLUDE file **xom1.h**. **tarheap** itself is only used the memory manager routine, **MEMLOW**.

For use on smaller computers (e.g., IBM-PCs) the size of **tarheap** can be significantly reduced to 500,000 8 byte words (4 megabytes), which is still large enough to run the vast majority of TART95 problems, and allows TART95 to be run on computers with as little as 8 megabytes of memory.

The size of the array **tarheap** is controlled by the parameter **MAXCORE** in the include file **xom1.h**. As distributed **MAXCORE** = 4,000,000 (4,000,000 REAL*8 words = 32 megabytes). For use on smaller computers it is merely necessary to reduce **MAXCORE** and re-compile the code. Note, when **MAXCORE** is changed, since it is only used in the memory manager routine **MEMLOW,** this is the only routine that MUST be re-compiled. **Warning** - do not reduce the size of **tarheap** to less than 500,000 words - for use on an 8 megabyte IBM-PC this will allow for: 4 megabytes for the system and the code, and the following 4 megabytes (500,000 words x 8 bytes per word = 4 megabytes) to be used by dynamic arrays.

Memory on Workstations

Workstations are so large these days that memory is no longer a restriction in implementing codes. Even the 32 megabytes used by TART95 is only a fraction of the memory available on modern workstations. The availability of these large workstations means that if you want to run problems that are much larger than those allowed by the standard distributed version of TART95 you need merely increase the parameter sizes defined in the include file **xom1.h** and re-compile the code. For example, the standard code allows up to 1,000 zones. If you want to allow 10,000 or even 100,000 zones merely change the parameter **MAXZONE**, re-compile the code and away you go.

Memory on IBM-PC

The memory size of PCs has also increased tremendously, but not to the degree that it has increased on workstations. Therefore memory size is still a consideration on PCs. Programs such as Windows and Smartdrive can be real memory hogs that use so much memory you may not be able to execute large programs, such as TART95. Below we will try to tell you how to make the most memory available for TART95 use; indeed the below remarks apply to any large code that you may have trouble running.

The 8, 16, and 32 megabyte versions on TART95 that are distributed will run on your PC ONLY if you free as much Extend Memory as possible. Run TART95 only under DOS, not under Windows, and you should turn off all caching programs, such as Smart Drive. You should edit your AUTOEXEC.BAT and CONFIG.SYS files to comment out caching routines and then restart your computer. At the DOS prompt type mem. This will give you a summary of how the memory of your computer is being used. Note, in particular how much of your extended memory is available and being used. If no extended memory is available you cannot run TART95. In this case you MUST turn on extended memory use; see your DOS documentation. If extended memory is available, on most systems about 500 to 600 kilobytes of it will be in use; this is acceptable. If megabytes of extended memory are being used you still have a caching program running; see your AUTOEXEC.BAT and CONFIG.SYS files.

If you have an 8 megabyte PC and all else fails, we should point out that these days PC memory costs about $ 35-40 a megabyte. So that upgrading from an 8 to 16 megabyte PC should cost you about $ 280-320. We strongly suggest that you consider upgrading the size of your PC. TART95 is nowhere near the largest code available for use on PCs, and with more memory you will not only be able to use a larger version of TART95, you will be able to use many more codes that simply cannot be used on smaller PCs.